COMP90042

Web search and text analysis

Workshop Week 10

Your tutor

- Winn Chow (Senior Tutor)
- winn.chow1@unimelb.edu.au
- Office: Doug McDonell - 9.23
- Here, you can find my workshop slides:
- https://github.com/winnchow/COMP90042-Workshops

Q1 and Q2

1. What is **chart parsing**? Why is it important?
2. Consider the following simple **context–free grammar**:

```
S -> NP VP
VP -> V NP | V NP PP
PP -> P NP
V -> "saw" | "walked"
NP -> "John" | "Bob" | Det N | Det N PP
Det -> "a" | "an" | "the" | "my"
N -> "man" | "cat" | "telescope" | "park"
P -> "on" | "by" | "with"
```

   (a) What changes need to be made to the grammar to make it suitable for **CYK parsing**?

   (b) Using the CYK strategy and the above grammar in CNF, parse the following sentences:

      i. "a man saw John"
      ii. "an park by Bob walked an park with Bob"
      iii. "park by the cat with my telescope"

Q1

– Chart parsing

COMP90042 W.S.T.A. (S1 2019) L17

Parsing CFGs

- Parsing: given string, identify possible structures
- Brute force search is intractable for non-trivial grammars
  - Good solutions use dynamic programming

COMP90042 W.S.T.A. (S1 2019)

S → NP VP
NP → DT NN
VP → VBD NP
DT → *the*
NN → *rat*
NN → *cheese*
VBD → *ate*

the	*rat*	*ate*	*the*	*cheese*
DT [0,1]	NP [0,2]	[0,3]	[0,4]	S [0,5]
	NN [1,2]	[1,3]	[1,4]	[1,5]
		VBD [2,3]	[2,4]	VP [2,5]
			DT [3,4]	NP [3,5]
				NN [4,5]

CYK by example

Q2a

- **Productions** (rules)

  W → X Y Z

  * Exactly one non-terminal on left-hand side (LHS)
  * An ordered list of symbols on right-hand side (RHS)
    — can be **Terminals** or **Non-terminals**

– Context Free Grammar

– CYK parsing:

  – Convert grammar to Chomsky Normal Form (CNF)

- Change grammar so all rules of form

  A → B C or A → a

```
S -> NP VP
VP -> V NP | V NP PP
PP -> P NP
V -> "saw" | "walked"
NP -> "John" | "Bob" | Det N | Det N PP
Det -> "a" | "an" | "the" | "my"
N -> "man" | "cat" | "telescope" | "park"
P -> "on" | "by" | "with"
```

Q2a

- CYK parsing:
  - Convert grammar to Chomsky Normal Form (CNF)

```
S -> NP VP
VP -> V NP | V NP PP
PP -> P NP
V -> "saw" | "walked"
NP -> "John" | "Bob" | Det N | Det N PP
Det -> "a" | "an" | "the" | "my"
N -> "man" | "cat" | "telescope" | "park"
P -> "on" | "by" | "with"
```

VP -> V X
X -> NP PP

NP -> Det Y
Y -> N PP

Q2b (i) “a man saw John”

	0	1	2	3	4
	a	man	saw	John	
0	[0,1]	[0,2]	[0,3]	[0,4]	
1		[1,2]	[1,3]	[1,4]	
2			[2,3]	[2,4]	
3				[3,4]	

```
S -> NP VP
VP -> V NP | V NP PP
PP -> P NP
V -> "saw" | "walked"
NP -> "John" | "Bob" | Det N | Det N PP
Det -> "a" | "an" | "the" | "my"
N -> "man" | "cat" | "telescope" | "park"
P -> "on" | "by" | "with"
```

VP -> V X
X -> NP PP

NP -> Det Y
Y -> N PP

Q2b (i) “a man saw John”

	0	1	2	3	4
	a	man	saw	John	
0	[0,1] Det	[0,2] NP	[0,3]	[0,4]	
1		[1,2] N	[1,3]	[1,4]	
2			[2,3]	[2,4]	
3				[3,4]	

```
S -> NP VP
VP -> V NP | V NP PP
PP -> P NP
V -> "saw" | "walked"
NP -> "John" | "Bob" | Det N | Det N PP
Det -> "a" | "an" | "the" | "my"
N -> "man" | "cat" | "telescope" | "park"
P -> "on" | "by" | "with"
```

VP -> V X
X -> NP PP

NP -> Det Y
Y -> N PP

Q2b (i) “a man saw John”

	0	1	2	3	4
	a	man	saw	John	
0	[0,1] Det	[0,2] NP	[0,3] -	[0,4]	
1		[1,2] N	[1,3] -	[1,4]	
2			[2,3] V	[2,4]	
3				[3,4]	

```
S -> NP VP
VP -> V NP | V NP PP
PP -> P NP
V -> "saw" | "walked"
NP -> "John" | "Bob" | Det N | Det N PP
Det -> "a" | "an" | "the" | "my"
N -> "man" | "cat" | "telescope" | "park"
P -> "on" | "by" | "with"
```

VP -> V X
X -> NP PP

NP -> Det Y
Y -> N PP

Q2b (i) “a man saw John”

	0	1	2	3	4
	a	man	saw	John	
0	[0,1] Det	[0,2] NP	[0,3] -	[0,4] S	
1		[1,2] N	[1,3] -	[1,4]	
2			[2,3] V	[2,4] VP	
3				[3,4] NP	

```
S -> NP VP
VP -> V NP | V NP PP
PP -> P NP
V -> "saw" | "walked"
NP -> "John" | "Bob" | Det N | Det N PP
Det -> "a" | "an" | "the" | "my"
N -> "man" | "cat" | "telescope" | "park"
P -> "on" | "by" | "with"
```

VP -> V X
X -> NP PP

NP -> Det Y
Y -> N PP

Q2b (ii) “an park by Bob walked an park with Bob”

an	*park*	*by*	*Bob*	*walked*	*an*	*park*	*with*	*Bob*
[0,1] Det	[0,2] NP	[0,3] -	[0,4] NP, X	[0,5] -	[0,6] -	[0,7] S	[0,8] -	[0,9] S, S
	[1,2] N	[1,3] -	[1,4] Y	[1,5] -	[1,6] -	[1,7] -	[1,8] -	[1,9] -
		[2,3] P	[2,4] PP	[2,5] -	[2,6] -	[2,7] -	[2,8] -	[2,9] -
			[3,4] NP	[3,5] -	[3,6] -	[3,7] S	[3,8] -	[3,9] S, S
				[4,5] V	[4,6] -	[4,7] VP	[4,8] -	[4,9] VP, VP
					[5,6] Det	[5,7] NP	[5,8] -	[5,9] NP, X
						[6,7] N	[6,8] -	[6,9] Y
							[7,8] P	[7,9] PP
								[8,9] NP

Q2b (iii) “park by the cat with my telescope”

park	*by*	*the*	*cat*	*with*	*my*	*telescope*
[0,1] N	[0,2] -	[0,3] -	[0,4] Y	[0,5] -	[0,6] -	[0,7] Y
	[1,2] P	[1,3] -	[1,4] PP	[1,5] -	[1,6] -	[1,7] PP
		[2,3] Det	[2,4] NP	[2,5] -	[2,6] -	[2,7] NP, X
			[3,4] N	[3,5] -	[3,6] -	[3,7] Y
				[4,5] P	[4,6] -	[4,7] PP
					[5,6] Det	[5,7] NP
						[6,7] N

Q3 and Q4

3. What differentiates **probabilistic parsing** from **chart parsing**? Why is this important? How does this affect the algorithms used for parsing?
4. What is a **probabilistic grammar** and what problem does it attempt to solve?

Q3

– Probabilistic Parsing

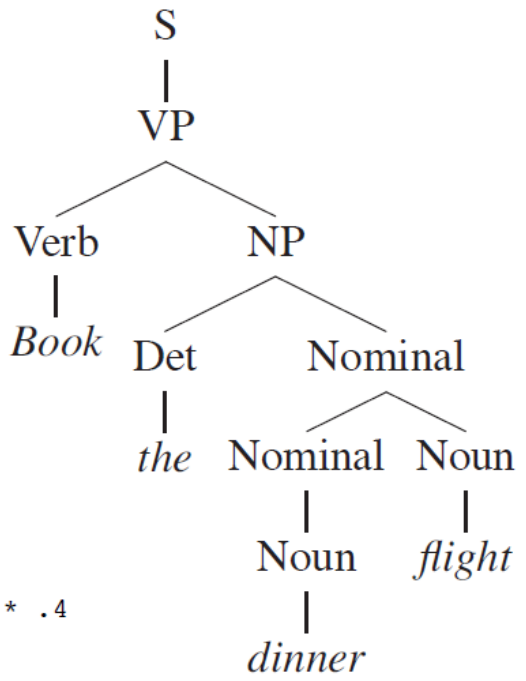
- In addition, store a **probability** with each production
  - NP → DT NN [p = 0.45]
  - NN → cat [p = 0.02]
  - NN → leprechaun [p = 0.00001]
  - …

How likely is a tree?

- Given a tree, we can compute its probability
  - Decomposes into probability of each production
- E.g., for tree on right,
  - P(tree) =

P(S → VP) ×
P(VP → Verb NP) ×
P(Verb → *Book*) ×
P(NP → Det Nominal) ×
P(Det → *the*) ×
P(Nominal → Nominal Noun) ×
P(Nominal → Noun) ×
P(Noun → *dinner*) ×
P(Noun → *flight*) = 2.16×10^{-6}

I.e., .05 * .2 * .3 * .2 * .6 * .2 * .75 * .1 * .4



Q4

– Probabilistic Grammar

- A **probabilistic grammar**, as described above, adds real-valued weights (probability) to each production in the grammar. This attempts to provide a "language model", that is, describe the likely sentences in a language, which is facilitated by their grammatical structure.

Q5

5. A hidden Markov model assigns each word in a sentence with a tag, e.g.,

   Donald/NNP has/VBZ small/JJ hands/NNS

   The probability of the sequence is based on the tag-word pairs, and the pairs of adjacent tags. Show how this process can be framed as a CFG, and how the various probabilities (e.g., observation, transition, and initial state) can be assigned to productions. What are the similarities and differences between CYK parsing with this grammar, and the HMM's Viterbi algorithm for finding the best scoring state sequence?

Q5

Donald/NNP has/VBZ small/JJ hands/NNS

HMM

<s>

|

NNP – Donald

|

VBZ – has

|

JJ – small

|

NNS – hands

Transition	Emission
<s> -> NNP NNP’	NNP -> Donald
NNP’ -> VBZ VBZ’	VBZ -> has
VBZ’ -> JJ JJ’	JJ -> small
JJ’ -> NNS	NNS -> hands